PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant/ Appellant:	Kireeti Kompella	Confirmation No.	9695
Serial No.:	10/045,717		
Filed:	October 19, 2001	Customer No.:	28863
Examiner:	Peling Andy Shaw		
Group Art Unit:	2144		
Docket No.:	1014-013US01/JNP-0082		
Title:	NETWORK ROUTING USING INDIRECT NEXT HOP DATA		

CERTIFICATE UNDER 37 CFR 1.8 I hereby certify that this correspondence is being transmitted via the United States Patent and Trademark Office electronic filing system on November 21, 2007

By: Caryl Harriman
Name: Caryl Harriman

REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450,
Alexandria, VA 22313

Dear Sir:

This is a Reply Brief responsive to the Examiner's Answer mailed September 25, 2007. The Appeal Brief was filed on June 11, 2007. The Notice of Appeal was filed on April 11, 2007 from the final Office Action mailed January 12, 2007.

No fee is believed due. Please charge any deficiencies or credits to Deposit Account No. 50-1778.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The Examiner accepted Appellant's statement in the Appeal Brief.

RELATED APPEALS AND INTERFERENCES

The Examiner accepted Appellant's statement in the Appeal Brief.

STATUS OF CLAIMS

The Examiner accepted Appellant's statement in the Appeal Brief.

STATUS OF AMENDMENTS

The Examiner accepted Appellant's statement in the Appeal Brief.

SUMMARY OF CLAIMED SUBJECT MATTER

The Examiner accepted Appellant's statement in the Appeal Brief.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The Examiner accepted Appellant's statement in the Appeal Brief.

ARGUMENT

In response to the Examiner's Answer, Appellant requests consideration of the following arguments, which supplement the arguments presented in the Appeal Brief.

In the "Grounds of Rejection" section of the Examiner's Answer, the Examiner appears to have advanced the same arguments and rejections presented in the final Office Action. Appellant directs the Board of Patent Appeals to Appellant's original Appeal Brief, which addresses the rejections.

The following discussion is responsive to the "Response to Argument" section of the Examiner's Answer.

FIRST GROUND OF REJECTION UNDER APPEAL

(Claims 1-2, 5-6, 12-14, 16-18, 22-23, 28-31, 37 and 40)

The Examiner maintains that the features of independent claim 1 are taught in Aramaki. In order to support this conclusion, the Examiner cites to passages of Aramaki that do not even concern forwarding tree-based routing schemes. As noted in the original Appeal Brief, Aramaki describes so-called Expansion Methods as table-based alternatives to forwarding tree-based routing methods. The Background section of Aramaki and FIGS. 1 and 2 make it clear that forwarding tree-based routing methods described by Aramaki for background purposes are different than such table-based alternatives shown in FIG. 3 of Aramaki. Therefore, the Examiner's conclusions that the tables of Aramaki are somehow a forwarding tree are contrary to the teaching of Aramaki.

Claim 1 recites (among other things) storing, external to a forwarding tree, next hop data representing network devices neighboring the network router, and storing, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to next hop data, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device. This allows for simple changes to the next hop data (external to the forwarding tree) to impact many leaf nodes of the forwarding tree without needing new leaf node destinations to be re-calculated or re-resolved.

In the discussion below, Appellant attempts to address the Examiner's various interpretations of Aramaki, which Appellant believes to be in direct conflict with the actual teaching of Aramaki. In many cases, however, Appellant has great difficulty in understanding the Examiner's positions.

Fundamentally, even if the tables utilized in the Aramaki approach could be construed as being some kind of forwarding tree (as required by the Examiner's analysis), the basic concept Appellant's claims is still lacking from Aramaki. In particular, Aramaki describes no data structure that stores indirect next hop data in leaf nodes of a forwarding tree, where the indirect next hop data maps the leaf nodes to next hop data that is stored external to the forwarding tree. These technical features are absent from Aramaki and, as a result, Aramaki fails to address the problem solved by Applicant's approach.

Specifically, the ultimate result in Aramaki upon parsing through the bits of a routing key when forwarding a packet is that a "hop pointer [is] read from a finally accessed entry of the second table as a retrieval result." See Abstract of Aramaki. Aramaki makes clear that the hop pointer is "indicative of the forwarding destination." See column 9, lines 54-56 of Aramaki. Thus, the hop pointer stored in the Aramaki reference is itself indicative of the forwarding destination. In no way does this hop pointer in Aramaki define a mapping to data external to the forwarding tree that indicates the next hop, but instead, in Aramaki, the forwarding destination itself is the finally accessed entry.

Even more specifically, Aramaki indicates that the hop pointer (i.e., the finally accessed entry of the second table that defines the retrieval result) indicates an IP address. See Column 16, lines 1-2 of Aramaki. Accordingly, Aramaki describes no data structure that stores indirect next hop data in leaf nodes of a forwarding tree, where the indirect next hop data maps the leaf nodes to next hop data that is stored external to the forwarding tree.

The table-based approach of Aramaki still requires that next hop data (i.e., a hop pointer) is stored within the tables themselves. This is fundamentally different than the idea reflected in claim 1 of storing indirect next hop data in leaf nodes of a forwarding tree, which map to next hop data stored external to the forwarding tree. Appellant's

claimed invention allows for changes to the next hop data external to the forwarding tree to impact the indirect next hop data of the forwarding tree without requiring re-calculation of the leaf nodes of the forwarding tree. In direct conflict with this easy updating process supported by Appellant's claimed invention, Aramaki specifically describes a more complex hop pointer updating process, as well as a process for changing a retrieval table (the second table) that requires changes of first table pointers and hop pointers. See columns 15 and 16 of Aramaki.

1.

In section 1 of the Examiner's Response to Arguments on the First Ground of Rejection, the Examiner summarized one of Appellant's arguments as follows:

> Appellant has argued (2nd paragraph on page 12) nothing in Aramaki teaches or suggests storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network router.

The Examiner then provided discussion of Aramaki before concluding that "it is clear that Aramaki does have the limitation of 'next hop data representing network devices neighboring the network router'."

As noted above, Appellant has difficulty understanding the Examiner's position. As best as Appellant's can discern, the Examiner's argument appears to be that the first tables of Aramaki are construed as a forwarding tree, and the second table is construed as containing next hop data that is external to the forwarding tree. This conclusion is erroneous insofar as the first tables are not a forwarding tree that defines leaf nodes. The actual teaching of Aramaki contradicts this interpretation advanced by the Examiner insofar as Aramaki clearly recognizes and distinguishes forwarding tree-based routing approaches from table-based approaches.

Appellant also notes that the Examiner's conclusion in section 1 of the Examiner's Answer that Aramaki "does have the limitation of next hop data representing network devices neighboring the network router" does not mean that Aramaki teaches or suggests storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network

router, which is the limitation of Appellant's claims that is addressed in section 1 of the Examiner's Answer.

Aramaki fails to disclose or suggest storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data representing network devices neighboring the network router. As explained in the Appeal Brief, Aramaki is not related to packet forwarding scheme that uses a forwarding tree. Instead, the teaching of Aramaki describes a packet forwarding scheme that uses sets of routing tables to be used instead of forwarding-tree based schemes. While Aramaki mentions radix tree retrieval methods in the Background section as relevant prior art for routing schemes, the specific features of Appellant's claims, which require storing indirect next hop data in leaf nodes of a forwarding tree and next hop data external to the forwarding tree, are not suggested by Aramaki.

As explained in the Appeal Brief, Aramaki distinguishes radix tree retrieval methods from the "expansion methods" that are based on retrieval tables as described in the Detailed Description of Aramaki. To be sure, the Background of Aramaki clearly distinguishes these different methods, first mentioning binary tree retrieval methods (col. 2, lines 5- 21), next explaining radix tree methods (col. 2, lines 22-51), and then explaining conventional expansion methods that use retrieval tables (beginning on col. 2, lines 52) that form the basis for the Aramaki invention.

FIGS. 1 and 2 of Aramaki clearly show that forwarding tree routing methods are different than the table-based approaches like that shown in FIG. 3 of Aramaki. Therefore, the Examiner's basic conclusions that the tables of Aramaki are themselves some type of forwarding tree is directly in conflict with the teaching of Aramaki, which recognizes that table-based routing schemes are different than forwarding tree-based routing schemes.

In the Examiner's Answer, the Examiner once again cited column 5, lines 48-67 of Aramaki in support of the rejections, apparently concluding that column 5, lines 48-67 of Aramaki suggests storing a forwarding tree having a set of nodes, and storing, external to the forwarding tree, next hop data. The teaching of Aramaki at column 5, lines 48-67, however, does not concern any use of forwarding tree. Instead, the teaching of Aramaki at column 5, lines 48-67 describes a table-based routing scheme that makes use of first

tables and a second table. The second table serves as an index table of the first tables to help limit the amount of data that needs to be stored to define network routes. None of the tables of Aramaki are a forwarding tree that includes leaf nodes. Instead, these tables are used as routing tables within a routing scheme based on the so-called "expansion methods" discussed in Aramaki. As specifically set forth in Aramaki's Background, such techniques are alternatives forwarding tree techniques.

In the Examiner's Answer, the Examiner stated that:

> Aramaki has shown (citing column 5, lines 46-67) that first tables are hierarchically arranged according to division of destination address, i.e., nodes in levels (hierarchs) per routing destination addresses; a second table indexes into the first tables, i.e., providing pointer among levels of nodes; first table contains pointers to the entries in the second table; traverses through first tables unit a hop pointer, i.e., indirect hop data" (sic) is reached in the second table.

The Examiner then stated that:

> hierarchical nodes (entries in first tables) are connected with pointers (entries in second table); a forwarding (routing) tree is established and next hop data (hop pointer) is in the second table.

Appellant is having difficulty understanding the Examiner's position, but maintains that the tables of Aramaki are not a forwarding tree. Instead, the first tables and second tables collectively form an alternative routing mechanism to a forwarding tree. Furthermore, the Examiner has failed to show any next hop data that is stored external to the forwarding tree.

As best as Appellant's can discern, the Examiner's argument appears to be that the first tables of Aramaki define a forwarding tree that includes leaf nodes, and the second table contains next hop data that is external to the forwarding tree. This conclusion is incorrect insofar as the first tables are not a forwarding tree, and do not include leaf nodes as required by Appellant's claim 1.

Furthermore, the Examiner's interpretation of the first tables of Aramaki as defining a forwarding tree that includes leaf nodes contradicts other interpretations of Aramaki that the Examiner advances in the Examiner's answer. These contradictions are addressed below.

Fundamentally, even if the tables of Aramaki could be construed as being some kind of forwarding tree (as provided in the Examiner's analysis), the basic concept Appellant's claims is still lacking from Aramaki. In particular, Aramaki does not store indirect next hop data in leaf nodes of a forwarding tree, which map to next hop data stored external to the forwarding tree.

On the contrary, the ultimate result in Aramaki upon parsing through the bits of a routing key is a "hop pointer read from a finally accessed entry of the second table as a retrieval result." See Abstract of Aramaki. Again, Aramaki makes clear that the hop pointer is "indicative of the forwarding destination." See column 9, lines 54-56 of Aramaki, and that the hop pointer (i.e., the finally accessed entry of the second table that defines the retrieval result) indicates an IP address. See Column 16, lines 1-2 of Aramaki. Thus, the hop pointer stored in the Aramaki reference is itself an IP address indicative of the forwarding destination.

This is fundamentally different than the idea reflected in claim 1 of storing indirect next hop data in leaf nodes of a forwarding tree, which map to next hop data stored external to the forwarding tree. Appellant's claimed invention allows for changes to the next hop data external to the forwarding tree to impact the indirect next hop data within the forwarding tree without requiring re-calculation of the forwarding tree or re-calculation of the indirect next hop data. In direct conflict with this easy updating process supported by Appellant's claimed invention, Aramaki specifically describes a more complex hop pointer updating process, as well as a process for changing a retrieval table (the second table) that requires changes of first table pointers and hop pointers. See columns 15 and 16 of Aramaki.

2.

In section 2 of the Examiner's Answer on the First Ground of Rejection, the Examiner addressed Appellant's claim limitation that requires "identifying a key within a network packet, and traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree."

The Examiner stated that:

> As one skilled in the art can see the first tables per FIG. 7 of Aramaki are used in traversing IP address bits (31-24, (23-16), (15-8) and (7-0) in a bitwise matching process through first tables for determination of next tree level pointer, i.e., second table entry, stored in first tables or a hop pointer.

Again, Appellant has difficulty understanding the Examiner's position. As best as Appellant's can discern, the Examiner is attributing the first tables of Aramaki to a forwarding tree that includes leaf nodes, and attributing the second table of Aramaki to next hop data that is external to the forwarding tree.

The bitwise matching process that the Examiner attributes to Aramaki, however, accesses both the first and second tables. This makes it clear that the first tables are not a forwarding tree that includes leaf nodes, as interpreted by the Examiner.

In other words, the Examiner's interpretation that the first table of Aramaki defines a forwarding tree that includes leaf nodes contradicts the Examiner's interpretation that the bitwise matching process of Aramaki "identifies a key within a network packet, and traverses a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree.

If the bitwise matching process of Aramaki actually traversed along a forwarding tree until reaching one of the leaf nodes of the forwarding tree, then the bitwise matching process would not access the second tables to arrive at the hop pointer, since the Examiner's interpretation requires the first tables to be the forwarding tree. The ultimate hop pointer read from the tables of Aramaki, however, is stored in the second table, and determined by traversing bits of a routing key through the first and second tables. Accordingly, the Examiner's interpretations of the first tables of Aramaki being a forwarding tree and the bitwise matching process of Aramaki being a traversal through the forwarding tree are inapposite insofar as an ultimate hop pointer read from the tables of Aramaki is in the second table, not the first table.

Furthermore, the ultimate hop pointer read from the tables of Aramaki is just that, a hop pointer. The entire concept of traversing a forwarding tree to leaf nodes that

include *indirect next hop data*, which maps to *next hop data* external to the forwarding tree is lacking from Aramaki.

3.

In section 3 of the Examiner's Answer on the First Ground of Rejection, the Examiner addressed Appellant's claim limitation that requires "at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring device."

In this analysis, the Examiner appears to confuse the concept of a routing key with the concept of leaf nodes of a forwarding table. The Examiner's contention appears to be that two different routing keys can ultimately point to the same next hop. Based on this, the Examiner seems to think that Aramaki suggests at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data (external to the forwarding tree) for the same neighboring device.

The Examiner is correct that conventional routing keys can ultimately point to the same next hop. That is to say, the ultimate hop pointer read from the second table in Aramaki could be the same for two or more different routing keys.

The fact remains, however, that the ultimate hop pointer read from the second table in Aramaki is not indirect next hop data that maps to external next hop data. The indirect next hop data according to Appellant's claimed invention serves as a map to a next hop (stored externally to the forwarding tree), allowing such next hops to be changed without requiring computationally intensive re-calculation of the leaf nodes of the forwarding tree.

4.

In section 4 of the Examiner's Answer on the First Ground of Rejection, the Examiner addressed Appellant's claim limitation that requires "upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree."

Appellant does not understand the Examiner's argument on these claim features. Aramaki clearly lacks these requirements of Appellant's claims.

In the table-based approach described by Aramaki (which again, does not use forwarding tables), the ultimate traversed path determined from bits of a routing key is "hop pointer read from a finally accessed entry of the second table." In contrast, the ultimate traversed path determined from bits of a routing key, according to Appellant's claimed invention, is indirect next hop data stored in a leaf node of a forwarding tree. The indirect next hop data in the leaf node, in turn, maps to next hop data that is external to the forwarding tree. Accordingly, Appellant's claimed invention can support changes to the next hop data without any recalculation of the forwarding tree that are traversed using the routing key.

Again, fundamentally, even if the tables of Aramaki could be construed as being some kind of forwarding tree (as provided in the Examiner's analysis), the basic concept Appellant's claims is still lacking from Aramaki. In particular, Aramaki does not store indirect next hop data in leaf nodes of a forwarding tree, which map to next hop data stored external to the forwarding tree.

On the contrary, the ultimate result in Aramaki upon parsing through the bits of a routing key is a "hop pointer read from a finally accessed entry of the second table as a retrieval result" and the hop pointer is described by Aramaki as being an IP address. This is fundamentally different than the idea reflected in claim 1 of storing indirect next hop data in leaf nodes of a forwarding tree, which map to next hop data stored external to the forwarding tree.

SECOND GROUND OF REJECTION UNDER APPEAL

(Claims 3-4, 7-11, 15, 19-21, 24-27, 32-36, 39 and 41-44)

1.

In section 1 of the Examiner's Answer on the Second Ground of Rejection, the Examiner addressed Appellant's arguments that Cain discusses routing techniques that specify primary and backup *routes*, not primary band backup *next hops*. A route specifies a beginning and an ending node, as well as every intermediate node along the route. A next hop, in contrast, is simply data representing a neighboring network device.

The Examiner's Answer on this point does not address Appellant's argument, and fails to demonstrate that Cain suggests storing, within each of the leaf nodes, a first

reference to a primary next hop within the next hop data external to the forwarding tree, and storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

In particular, the Examiner concludes that Cain shows routing through the same network interface where there are two different next-hop devices. This conclusion is irrelevant to the issue at hand, and fails to establish that the prior art suggests storing, within each of the leaf nodes, a first reference to a primary next hop within the next hop data external to the forwarding tree, and storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

Routing techniques that specify primary and backup routes do not suggest storing, within each of the leaf nodes of a forwarding tree, a first reference to a primary next hop within the next hop data external to the forwarding tree, and storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

2.

In section 2 of the Examiner's Answer on the Second Ground of Rejection, the Examiner addressed Appellant's arguments that nothing in Cain suggests receiving a packet comprising network update information, and modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

The Examiner did not cite any evidence, but simply concluded that:

> Cain has shown being able to prioritizing (sic) a preferred route and alternate route associated with different next-hop devices but over the same network interface. Thus, the forwarding-tree is not changed, but the information on next-hop devices on the same network interface."

Appellant submits that the Examiner has failed to even address the limitations of claims 7, 32 and 41. Even if the Examiner's statement above were correct, the statement fails to even contend that the prior art teaches receiving a packet comprising network update information, and modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

Appellant also submits (once again) that Cain teaches the antithesis of claims 7, 32 and 41 insofar as Cain teaches modification of network routes. Modification of network routes (if a forwarding tree were used) would comprise modification of a forwarding tree rather than modification to next hop data without any modification of a forwarding tree, as recited in Appellant's claims 7, 32 and 41.

On the contrary, in Cain, if a network update message is received, then the routes need to be recalculated. One specific advantage of Appellant's claimed invention is that such route recalculation can be avoided. In contrast to route re-calculation, per Cain, claims 7, 32 and 41 recite modification to next hop data (which resides external to a forwarding tree) without any modification of the forwarding tree.

Indeed, in direct conflict with this easy update process recited in claims 7, 32 and 41, Aramaki specifically describes a more complex hop pointer updating process, as well as a process for changing a retrieval table (the second table) that requires changes of first table pointers and hop pointers. See columns 15 and 16 of Aramaki. Nothing in this section of Aramaki or elsewhere suggests receiving a packet comprising network update information, and modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

3.

In section 3 of the Examiner's Answer on the Second Ground of Rejection, the Examiner cites column 5, lines 29-67 of Cain as teaching routing information stored within a routing engine, wherein the routing information represents routes within a network, and route data, indirect next hop data and the next hop data stored within a packet forwarding engine. This cited passage simply does not contemplate any router architecture that separates routing information in a routing engine and route data in a packet forwarding engine.

4.

In the Appeal Brief, Appellant argued that Cain fails to disclose anything akin to a packet forwarding engine, much less a much less a packet forwarding engine that stores the route data, the indirect next hop data and the next hop data. Furthermore, contrary to

the Examiner's analysis of claims 9, 25, 34 and 43, Appellant argued that nothing in Cain suggests receiving a packet comprising network topology update information, updating the routing information within the routing engine, and issuing a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

In section 4 of the Examiner's Answer on the Second Ground of Rejection, the Examiner failed to address these arguments that Cain lacks the teaching attributed to this reference by the Examiner. The Examiner did not identify anything in Cain that teaches the reception of a packet comprising network topology update information, the update of the routing information within the routing engine, or the issuance of a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

CONCLUSION OF ARGUMENT

Appellant requests consideration of the foregoing arguments, which supplement the arguments presented in the Appeal Brief. Appellant also requests reversal of all pending rejections, and allowance of all pending claims.

Respectfully submitted,

Date: November 21, 2007

Shumaker and Sieffert
1625 Radio Drive, Suite 300
St. Paul, Minnesota 55125
Telephone:(651) 286-8343
Facsimile: (651) 735-1102

By:

Name: Kelly Patrick Fitzgerald
Reg. No.: 46,326

CLAIMS APPENDIX:

Claim 1 (Previously presented): A method comprising:

storing, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network;

storing, external to the forwarding tree, next hop data representing network devices neighboring the network router;

storing, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device;

identifying a key within a network packet;

traversing a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree;

upon reaching a leaf node of the traversed path, using the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree; and

forwarding the packet to the selected next hop.

Claim 2 (Previously presented): The method of claim 1, wherein the forwarding tree comprises a radix tree.

Claim 3 (Previously presented): The method of claim 2, wherein storing the indirect next hop data comprises:

storing, within each of the leaf nodes, a first reference to a primary next hop within the next hop data external to the forwarding tree, and

storing, within each of the leaf nodes, a second reference to a backup next hop within the next hop data external to the forwarding tree.

Claim 4 (Original): The method of claim 3, further comprising routing packets to the backup next hop in response to a network event.

Claim 5 (Previously presented): The method of claim 2, wherein storing the indirect next hop data comprises storing a data pointer within each of the leaf nodes that references the next hop data external to the forwarding tree.

Claim 6 (Previously presented): The method of claim 1, wherein storing the next hop data comprises storing an array of next hop data elements external to the forwarding tree.

Claim 7 (Previously presented): The method of claim 1, further comprising:

receiving a packet comprising network update information; and

modifying the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

Claim 8 (Original): The method of claim 1, further comprising:

storing routing information within a routing engine, wherein the routing information represents routes within a network; and

storing the route data, the indirect next hop data and the next hop data within a packet forwarding engine.

Claim 9 (Previously presented): The method of claim 8, further comprising:

receiving a packet comprising network topology update information;

updating the routing information within the routing engine; and

issuing a message from the routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

Claim 10 (Original): The method of claim 8, wherein storing the routing information includes storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 11 (Original): The method of claim 9, wherein storing the routing information includes storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine, and issuing the message comprises analyzing the copy to identify the next hop for modification.

Claim 12 (Previously presented): A computer-readable medium having data structures therein that control forwarding of packets by a network device comprising:

a first data structure to store route data representing destinations within a computer network, wherein the first data structures is arranged as forwarding tree having a set of nodes, and wherein the nodes includes a set of leaf nodes that correspond to destinations within a computer network;

a second data structure external to the forwarding tree to store next hop data representing interfaces to neighboring network devices; and

a set of data structures, within the leaf nodes of the forwarding tree, to store indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data,

wherein the indirect next hop data causes the network device to, upon reaching a leaf node of a traversed path through the forwarding tree, select a next hop from the next hop data external to the forwarding tree and forward the packet to the selected next hop.

Claim 13 (Previously presented): The computer-readable medium of claim 12, wherein the forwarding tree comprises a radix tree.

Claim 14 (Original): The computer-readable medium of claim 12, wherein the indirect next hop data comprises a set of data pointers stored within the leaf nodes.

Claim 15 (Original): The computer-readable medium of claim 14, wherein the data pointers include pointers to primary next hops and pointers to backup next hops.

Claim 16 (Previously presented): The computer-readable medium of claim 12, wherein the second data structure comprises an array of next hop data elements.

Claim 17 (Previously presented): A router comprising:

a computer-readable medium to store: (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network, and , (ii) next hop data, external to the forwarding tree, representing neighboring network devices, and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data; and

a control unit that identifies a key within a network packet, traverses a path through the forwarding tree by testing bits of the key until reaching one of the leaf nodes of the forwarding tree,

wherein, upon reaching a leaf node of the traversed path, the control unit uses the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree and forwards the packet to the selected next hop.

Claim 18 (Original): The router of claim 17, wherein the indirect next hop data comprises a set of data pointers stored within the leaf nodes.

Claim 19 (Original): The router of claim 18, wherein the data pointers include pointers to primary next hops and pointers to backup next hops.

Claim 20 (Original): The router of claim 17, wherein some of the next hop data represents software modules for processing data packets.

Claim 21 (Original): The router of claim 20, wherein each of the software modules is selected from one of a packet filter, a policy enforcer and a packet counter.

Claim 22 (Previously presented): The router of claim 17, wherein the forwarding tree is arranged as a radix tree.

Claim 23 (Previously presented): The router of claim 22,

wherein the indirect next hop data includes a set of data pointers associated with the leaf nodes, and

wherein the data pointers reference portions of the next hop data stored external to the forwarding tree.

Claim 24 (Previously presented): A router comprising:

a routing engine to store routing information representing a topology of a network; and

a packet forwarding engine to store packet forwarding information in accordance with the routing information, the packet forwarding information including (i) a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network, and , , (ii) next hop data external to the forwarding tree, representing interfaces to neighboring network devices, and (iii) indirect next hop data, within the leaf nodes of the forwarding tree, that maps the leaf nodes of the forwarding tree to the next hop data.

Claim 25 (Previously presented): The router of claim 24, wherein the routing engine receives a packet comprising network topology update information and, in response to the network topology update information, updates the routing information and directs the packet forwarding engine to modify the next hop data.

Claim 26 (Original): The router of claim 24, wherein the routing information includes data structures storing a copy of the route data, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 27 (Original): The router of claim 26, wherein the routine engine analyzes the data structures to identify the next hop for modification.

Claim 28 (Previously presented): A computer-readable medium having instruction therein for causing a programmable processor within a router to:

store, within a network router, a forwarding tree having a set of nodes, wherein the nodes include leaf nodes that correspond to destinations within a computer network;

store, external to the forwarding tree, next hop data representing network devices neighboring the network router;

store, within the leaf nodes of the forwarding tree, indirect next hop data that map the leaf nodes of the forwarding tree to the next hop data, wherein at least two of the leaf nodes of the forwarding tree contain indirect next hop data that references the next hop data for the same neighboring network device;

identify a key within a network packet;

traverse a subset of the nodes of the forwarding tree within a network device by testing at least one bit of the key per each of the traversed nodes, wherein values of the tested bits in the key determine a path traversed along the forwarding tree until reaching one of the leaf nodes of the forwarding tree;

upon reaching a leaf node of the traversed path, use the indirect next hop data within the leaf node of the traversed path to select a next hop from the next hop data external to the forwarding tree; and

forward the packet to the selected next hop.

Claim 29 (Previously presented): The computer-readable medium of claim 28, wherein the instructions cause the processor to store the forwarding tree as a radix tree.

Claim 30 (Previously presented): The computer-readable medium of claim 29, wherein the instructions cause the processor to store the indirect next hop data as a respective data pointer within each of the leaf nodes,

wherein the data pointer within each of the leaf nodes reference the next hop data external to the forwarding tree.

Claim 31 (Original): The computer-readable medium of claim 28, wherein the instructions cause the processor to store an array of next hop data elements, and further wherein the portion of the next hop data comprises at least one next hop data elements.

Claim 32 (Previously presented): The computer-readable medium of claim 28, the instructions cause the processor to:

receive a packet comprising network update information; and

modify the next hop data in response to the network update information.

Claim 33 (Previously presented): The computer-readable medium of claim 28, the instructions cause the processor to:

store routing information within a routing engine, wherein the routing information represents routes within a network; and

store the forwarding tree, the indirect next hop data and the next hop data within a packet forwarding engine.

Claim 34 (Previously presented): The computer-readable medium of claim 33, the instructions cause the processor to:

receive a packet comprising network topology update information;

update the routing information within the routing engine; and

issue a message from the routing engine to direct the packet forwarding engine to modify the next hop data external to the forwarding tree in response to the network update information without modifying the forwarding tree.

Claim 35 (Previously presented): The computer-readable medium of claim 33, wherein the instructions cause the processor to store a copy of the forwarding tree, the indirect next hop data and the next hop data stored within the packet forwarding engine.

Claim 36 (Previously presented): The computer-readable medium of claim 33, wherein the instructions cause the processor to store a copy of the forwarding tree, the indirect next hop data and the next hop data stored within the packet forwarding engine, and issuing the message comprises analyzing the copy to identify the next hop for modification.

Claim 37 (Previously presented): A method comprising routing packets within a network using indirect next hop data that maps leaf nodes of a forwarding tree to next hop data stored external to the forwarding tree,

wherein the leaf nodes correspond to destinations within a computer network,

wherein the next hop data represents next hops within a network, and

wherein at least two different ones of the leaf nodes of the forwarding tree contain indirect next hop data that references a same next hop within the next hop data.

Claim 38 (Cancelled).

Claim 39 (Previously presented): The method of claim 37, further comprising storing the indirect next hop data within the leaf nodes as pointers to primary next hops and pointers to backup next hops.

Claim 40 (Previously presented): The method of claim 37, wherein the forwarding tree comprises a radix tree.

Claim 41 (Previously presented): The method of claim 37, further comprising:

receiving a packet comprising network update information; and

modifying the next hop data in response to the network update information.

Claim 42 (Original): The method of claim 37, further comprising storing the indirect next hop data within a packet forwarding engine.

Claim 43 (Previously presented): The method of claim 42, further comprising:

receiving a packet comprising network topology update information;

issuing a message from a routing engine to direct the packet forwarding engine to modify the next hop data in response to the network update information.

Claim 44 (Original): The method of claim 42, further comprising storing a copy of the indirect next hop data within a routing engine.

EVIDENCE APPENDIX:

NONE

RELATED PROCEEDINGS APPENDIX:

NONE